



# UTILIZAÇÃO DE REDES NEURAIS PARA CONSTRUÇÃO DA CURVA DE POTÊNCIA DE AEROGERAADORES

Trabalho de Conclusão de Curso  
Engenharia da Computação

**Artur Sivini de Siqueira**

**Orientador:** Prof. Dr. Mêuser Jorge Silva Valença



Artur Sivini de Siqueira

*UTILIZAÇÃO DE REDES NEURAIS PARA  
CONSTRUÇÃO DA CURVA DE POTÊNCIA  
DE AEROGERADROES*

Trabalho de Conclusão de Curso apresentado  
como requisito parcial para obtenção do di-  
ploma de Bacharel em Engenharia de Com-  
putação pela Escola Politécnica de Pernam-  
buco - Universidade de Pernambuco

Orientador:

Prof. Dr. Mêuser Jorge Silva Valença

UNIVERSIDADE DE PERNAMBUCO  
ESCOLA POLITÉCNICA DE PERNAMBUCO  
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Recife - PE, Brasil

28 de novembro de 2013

### MONOGRAFIA DE FINAL DE CURSO

#### Avaliação Final (para o presidente da banca)\*

No dia 17 de 12 de 2013, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente ARTUR SIVINI DE SIQUEIRA, orientado pelo professor Meuser Jorge Silva Valença, sob título Utilização de redes neurais para construção da curva de potência de aerogeradores, a banca composta pelos professores:

**Sérgio Murilo Maciel Fernandes**

**Meuser Jorge Silva Valença**

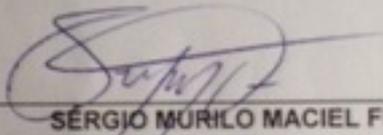
Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

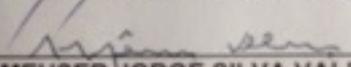
Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 10,0 ( DEZ )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 03 dias para entrega da versão final da monografia a contar da data deste documento.

  
SÉRGIO MURILO MACIEL FERNANDES

  
MEUSER JORGE SILVA VALENÇA

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

Dedico este trabalho aos meus pais, Jucenildo e Helenice, ao meu irmão, Filipe, à minha namorada, Alessandra, e à Deus.

# *Agradecimentos*

Agradeço, primeiramente, a Deus por permitir que tudo isso se concretizasse. Sem a Sua vontade não seria possível chegar ao fim do curso de graduação.

Agradeço também à minha família: meus pais, Jucenildo Siqueira e Helenice Sivini, e meu irmão Filipe Sivini. Agradeço pela força concedida e desculpo-me pelos momentos de ausência, momentos estes destinados a estudos, à realização de projetos e à elaboração deste trabalho.

Agradeço à minha namorada, Alessandra Soares. Por que não dizer que ela também foi uma professora para mim?! Sua ajuda foi imprescindível para a minha formação. Entretanto, agradeço, sobretudo, pela sua presença ao meu lado desde o primeiro dia de aula.

Agradeço também ao meu orientador, Prof. Dr. Mêuser Jorge Silva Valença. Ele que foi responsável por guiar o desenvolvimento deste trabalho: sempre auxiliando, corrigindo e aconselhando.

Agradeço, por fim, a todos os que colaboraram para minha formação acadêmica. Todos aqueles que contribuíram para que eu me tornasse um estudante, um profissional e uma pessoa melhor.

# *Resumo*

A energia elétrica se tornou um bem imprescindível para o homem hodierno. Existe uma tamanha dependência que é inimaginável o cotidiano da sociedade sem a utilização de energia. Entretanto, este consumo desenfreado ocasiona problemas de cunho ambiental: grande parte da energia elétrica provém de fontes de origem fóssil, as quais agravam o efeito estufa e não são renováveis. Em face destes problemas, vivencia-se, no cenário mundial, um momento de fortalecimento das fontes energéticas limpas e renováveis. Dentre elas, destaca-se a energia eólica, uma vez que se apresenta como alternativa pouco financeiramente custosa e de baixos impactos ambientais. Para que os parques eólicos, estruturas destinadas a transformar energia eólica em elétrica, operem de forma eficaz, faz-se necessário um bom planejamento embasado na estimativa da potência elétrica gerada em função da velocidade do vento. Portanto, este trabalho possui como objetivo a utilização de Redes Neurais Artificiais para a construção de curvas de potência de aerogeradores. Serão realizadas comparações entre os resultados obtidos a partir do emprego de redes *Multi-Layer Perceptron* e redes *Radial Basis Function*. Também serão efetuadas comparações com a função sigmoide e um polinômio de 2<sup>o</sup> grau. Por fim, a partir de testes estatísticos, será analisada qual técnica obteve melhor desempenho.

**Palavras-chave:** Energia Eólica, Curva de Potência, Redes Neurais Artificiais, *Multi-Layer Perceptron*, *Radial Basis Function*.

# *Abstract*

Electricity has become an essential property for today's man. There is such a huge dependency that is unimaginable to everyday society without using energy. However, this unbridled consumption, causes problems of an environmental nature: most of the electricity comes from fossil sources, which aggravate the greenhouse effect and are not renewable. In the face of these problems, one experiences on the world stage, a time of strengthening of clean and renewable energy sources. Among them, there is the wind, since it is an alternative bit financially costly and low environmental impact. For wind farms, structures designed to transform wind energy into electricity, to operate effectively, it is necessary to good planning grounded in the estimation of electrical power generated as a function of wind speed. Therefore, this work aims to use Artificial Neural Networks to construct power curves of wind turbines. Comparisons are made between the results obtained from the use of a Multi-Layer Perceptron network and Radial Basis Function network. Comparisons are also made with the sigmoid function and a polynomial of degree 2. Finally, from statistical tests will be analyzed which technique performed better.

**Keywords:** Wind Power, Wind Turbine Power Curve, Artificial Neural Networks, Multi-Layer Perceptron, Radial Basis Function.

# Sumário

<b>Lista de Figuras</b>	p. x
<b>Lista de Tabelas</b>	p. xi
<b>Lista de Abreviaturas e Siglas</b>	p. xi
<b>1 Introdução</b>	p. 12
1.1 Motivação . . . . .	p. 12
1.2 Objetivos . . . . .	p. 14
1.2.1 Objetivo Geral . . . . .	p. 14
1.2.2 Objetivos Específicos . . . . .	p. 14
1.3 Estrutura da Monografia . . . . .	p. 14
<b>2 Fundamentação Teórica</b>	p. 16
2.1 Energia Eólica . . . . .	p. 16
2.1.1 Introdução . . . . .	p. 16
2.1.2 Aerogeradores . . . . .	p. 17
2.1.3 Curva de Potência . . . . .	p. 19
2.1.4 Potencial Brasileiro de Geração de Energia Eólica . . . . .	p. 20
2.2 Redes Neurais Artificiais . . . . .	p. 20
2.3 Redes MLP . . . . .	p. 23
2.3.1 Treinamento com algoritmo <i>backpropagation</i> . . . . .	p. 23
2.3.2 Treinamento com algoritmo de Levenberg-Marquardt . . . . .	p. 24

2.4	Redes com Função de Base Radial . . . . .	p. 25
2.4.1	Treinamento de redes RBF . . . . .	p. 25
<b>3</b>	<b>Metodologia</b>	p. 28
3.1	Base de Dados . . . . .	p. 28
3.2	Pré-processamento dos Dados . . . . .	p. 28
3.3	Construção da Curva de Potência Utilizando Polinômio de 2 <sup>o</sup> Grau . . . . .	p. 29
3.3.1	Características da Implementação . . . . .	p. 29
3.4	Construção da Curva de Potência Utilizando Função Sigmóide . . . . .	p. 30
3.4.1	Características da Implementação . . . . .	p. 30
3.5	Construção da Curva de Potência Utilizando Redes MLP . . . . .	p. 31
3.5.1	Subdivisão dos Dados . . . . .	p. 31
3.5.2	Arquitetura das Redes . . . . .	p. 31
3.5.3	Treinamento . . . . .	p. 32
3.5.3.1	Critério de Parada . . . . .	p. 33
3.5.4	Características da Implementação . . . . .	p. 33
3.6	Construção da Curva de Potência Utilizando Redes RBF . . . . .	p. 35
3.6.1	Subdivisão dos Dados . . . . .	p. 35
3.6.2	Arquitetura da Rede . . . . .	p. 36
3.6.3	Características da Implementação . . . . .	p. 36
3.7	Execuções . . . . .	p. 37
3.8	Testes Estatísticos . . . . .	p. 37
<b>4</b>	<b>Resultados</b>	p. 39
4.1	Curva de Potência Construída com Polinômio do 2 <sup>o</sup> Grau . . . . .	p. 39
4.2	Curva de Potência Construída com Função Sigmóide . . . . .	p. 40

4.3	Curva de Potência Construída com redes MLP . . . . .	p. 41
4.4	Curva de Potência Construída com redes RBF . . . . .	p. 43
4.5	Comparações entre as Técnicas . . . . .	p. 45
<b>5</b>	<b>Considerações Finais</b>	p. 47
5.1	Conclusões . . . . .	p. 47
5.2	Trabalhos Futuros . . . . .	p. 48
	<b>Referências</b>	p. 49
	<b>Apêndice A – Resultados dos Testes Estatísticos Realizados no <i>Software</i></b>	
	<b>R</b>	p. 51
	<b>Apêndice B – Script dos Testes Estatísticos Realizados no <i>Software R</i></b>	p. 53

## *Lista de Figuras*

1	Parque eólico. . . . .	p. 13
2	Turbina de eixo vertical. . . . .	p. 18
3	Turbina de eixo horizontal. . . . .	p. 18
4	Curva de potência. . . . .	p. 19
5	Neurônio biológico. . . . .	p. 21
6	Neurônio proposto por McCulloch e Pitts. . . . .	p. 22
7	Curva de Potência Construída com Polinômio de 2 <sup>o</sup> Grau. . . . .	p. 40
8	Curva de Potência Construída com Função Sigmóide. . . . .	p. 41
9	Curva de Potência Construída com a Rede MLP do Experimento 8. . . . .	p. 43
10	Curva de Potência Construída com a Rede RBF do Experimento 2. . . . .	p. 45
11	Análise Estatística entre os Resultados do Polinômio de 2 <sup>o</sup> Grau e os da Função Sigmóide. . . . .	p. 51
12	Análise Estatística entre os Resultados da rede MLP e os da RBF. . . . .	p. 51
13	Análise Estatística entre os Resultados da rede MLP e os do Polinômio. . . . .	p. 52

## *Lista de Tabelas*

1	Resultados dos Experimentos com Polinômio de 2 <sup>o</sup> Grau . . . . .	p. 39
2	Resultados dos Experimentos com Função Sigmóide . . . . .	p. 40
3	Resultados dos Experimentos com Redes MLP . . . . .	p. 42
4	Resultados dos Experimentos com Redes RBF . . . . .	p. 44

## *Lista de Abreviaturas e Siglas*

AIE	<i>Agência Internacional de Energia</i>
CI	<i>Computação Inteligente</i>
EPMA	<i>Erro Percentual Médio Absoluto</i>
MLP	<i>Multilayer Perceptron</i>
RBF	<i>Radial Basis Function</i>

# 1 *Introdução*

Este capítulo é subdividido em três seções. Primeiramente, é exposta a motivação que engendrou este trabalho. Em seguida, são elencados os objetivos almejados. Por fim, são expostos a maneira segundo a qual os capítulos deste trabalho estão organizados e os assuntos neles abordados.

## 1.1 *Motivação*

A energia elétrica assume tamanha importância na sociedade hodierna que se pode afirmar: ao decair o fornecimento de energia, origina-se uma concomitante redução no volume financeiro em circulação. No Nordeste brasileiro, por exemplo, um apagão de cerca de 3 horas de duração acarretou prejuízos de R\$ 385 milhões (TAVARES, 2013).

O acontecimento mencionado retrata a inquestionável dependência do homem moderno para com a energia elétrica. Não é fácil pensar numa sociedade em que não seja possível ligar computador, ar-condicionado, lâmpada, chuveiro elétrico, marcapasso, etc. Entretanto, o consumo desmedido da energia ocasiona, no cenário mundial, problemas de cunho ambiental em razão de a grande maioria das fontes empregadas serem de origem fóssil.

A utilização de fontes como o petróleo, carvão mineral e gás natural possui duas grandes desvantagens. A primeira delas reside no fato de agravarem o efeito estufa, acarretando mudanças climáticas drásticas, poluição do ar, perda de espécies da fauna e flora (FREITAS, 2008). A segunda grande desvantagem consiste na característica de esses recursos naturais não serem renováveis. O processo de formação dos combustíveis fósseis dura milhões de anos, ou seja, quando são consumidos, não há meios de a natureza ou o homem repô-los num tempo hábil.

Em conformidade com dados da Agência Internacional de Energia (AIE), cerca de 81% da matriz energética mundial é proveniente de fontes fósseis (O GLOBO, 2013). Caso a média de consumo não seja reduzida, as reservas de petróleo e gás natural devem alcan-

çar o esgotamento em 100 anos e o carvão mineral, em 200 anos (CERQUEIRA, 2008).

Em razão dos problemas expostos, vivencia-se, no momento, um período de fortalecimento das fontes energéticas limpas e renováveis. Energia solar, eólica, maremotriz, biomassa surgem como alternativas proveitosas. Dentre essas fontes, destaca-se a energia eólica, visto que é uma das menos danosas ao meio ambiente e uma das menos financeiramente custosas (LUCIANA, 2013). Em relação ao âmbito nacional, o Brasil dispõe de grande potencial de geração de energia eólica devido ao fato de possuir um volume de ventos duas vezes maior do que a média mundial e, também, por apresentar baixa oscilação da velocidade do vento o que enseja uma maior previsibilidade da geração de energia elétrica (JANNUZZI, 2012).

A conversão de energia eólica em energia elétrica é realizada nos denominados parques eólicos, os quais são definidos como aglomerados de aerogeradores (Figura 1).



**Figura 1: Parque eólico.**

[Fonte: <http://www.noticiassin.com/>]

Para favorecer um funcionamento mais eficaz dos parques eólicos, faz-se necessário uma programação da geração diária, a qual depende da inconstância do vento. Portanto, é imprescindível a existência de modelos de previsão da geração eólica que considerem fatores como a velocidade do vento, sua direção e a potência elétrica que pode ser gerada.

Uma vez que o foco de um parque eólico reside no seu desempenho, a construção de uma curva de potência de seus aerogeradores se apresenta como algo altamente desejável e útil para prever a produção de energia elétrica do parque com base em características do vento (WAN; ELA; ORWIG, 2010). A curva de potência consiste num modelo que procura correlacionar a potência eólica gerada com a velocidade do vento.

Este trabalho propõe a utilização de Redes Neurais Artificiais (RNAs) para a construção de curvas de potência.

As RNAs compõem uma área da inteligência computacional bastante difundida cujo poderio reside na capacidade de aproximar inúmeras funções não lineares a partir de um mesmo modelo (RABUNAL; DORADO, 2006), donde surge a qualificação de aproximadora universal de funções. Portanto, as RNAs mostram-se, muitas vezes, ferramentas adequadas para a realização de previsões, classificações e reconhecimento de padrões.

## 1.2 **Objetivos**

### 1.2.1 **Objetivo Geral**

Analisar a aplicação de redes neurais artificiais como técnica de construção de curvas de potência de turbinas eólicas é o principal objetivo deste trabalho. Tal análise será embasada na comparação entre os resultados obtidos a partir da utilização de diferentes tipos de RNAs e os resultados provenientes da aplicação de uma função sigmóide e de um polinômio de 2° grau.

### 1.2.2 **Objetivos Específicos**

Tem-se como objetivos específicos:

1. Examinar a influência da velocidade do vento num determinado tempo 't', realizando experimentos com variações das entradas da rede com os valores de velocidade em 't - 1', 't - 2', ..., 't - 5';
2. Investigar se a inclusão da direção do vento como parâmetro de entrada da rede agrega valor à construção da curva de potência.

## 1.3 **Estrutura da Monografia**

No capítulo 2, explica-se a fundamentação teórica a partir da qual este trabalho foi desenvolvido, tais como os conceitos de energia eólica e de redes neurais artificiais, com ênfase nas arquiteturas utilizadas neste trabalho de conclusão de curso. O capítulo 3 aborda a metodologia empregada na persecução dos objetivos deste estudo, ou seja, o passo-a-passo realizado desde o pré-processamento dos dados até os testes estatísticos.

Na sequência, o capítulo 4 apresenta os resultados provenientes das simulações das técnicas escolhidas. Por último, o capítulo 5 sumariza o trabalho, expondo as dificuldades enfrentadas, as conclusões alcançadas e a sugestão de trabalhos futuros.

## 2 *Fundamentação Teórica*

Este capítulo apresenta conceitos e noções acerca da base teórica sobre a qual o trabalho foi desenvolvido. A seção 2.1 trata de conceitos base no cenário da energia eólica. Na seção 2.2, são abordados fundamentos das RNAs, técnica da Computação Inteligente (CI) bastante difundida cuja aplicação reside, sobretudo, em problemas de classificação, previsão e reconhecimento de padrões. As seções 2.3 e 2.4 tratam das arquiteturas de RNAs empregadas neste trabalho, sendo, respectivamente, a Rede Perceptron de Múltiplas Camadas (do inglês, *Multilayer Perceptron*, MLP) e a Rede com Função de Base Radial (do inglês, *Radial Basis Function*, RBF).

### 2.1 Energia Eólica

#### 2.1.1 Introdução

A energia eólica, apesar de ainda não ser amplamente utilizada, tem uma procedência bastante remota. Por volta de 200 A.C., os persas já empregavam moinhos de vento para bombeamento de água e moagem de grãos. Porém, apenas no fim do século XIX, foi criado, por Charles F. Brush, o primeiro cata-vento destinado à geração de energia elétrica (DUTRA, 2008).

A energia proveniente dos ventos tem, por sua vez, origem na energia solar. Segundo Rossi e Oliveira (2002), aquela é uma forma de energia gerada a partir do aquecimento diferenciado das camadas de ar, acontecimento que resulta na variação dos gradientes de pressão. À medida em que o ar é aquecido, ele se torna menos denso e sobe. O ar adjacente desloca-se, então, para a região de menor pressão. A esta movimentação do ar, atribui-se o nome: vento.

A disponibilidade de energia eólica está sujeita a muitas variações ao longo do ano. A razão disso está no fato de o regime dos ventos sofrer influência de diversos fatores. Portanto, antes da instalação de parques eólicos, faz-se necessário o estudo de componentes

como: a topografia e rugosidade da região; pressão atmosférica; densidade do ar; umidade relativa; temperatura.

### 2.1.2 Aeroogeradores

Também denominados turbinas eólicas, os aeroogeradores são dispositivos capazes de converter a energia cinética presente no vento em eletricidade. A quantidade de energia convertida depende, sobretudo, da velocidade do vento e do diâmetro do rotor (ROSSI; OLIVEIRA, 2002).

O vento, por sua vez, apresenta características que ditam a quantidade de energia que ele contém. A equação 2.1 calcula a chamada densidade de potência do vento, que representa a potência disponível em cada  $m^2$  do vento (LI et al., 2001):

$$P_w = 0.5\rho V^3 \quad (2.1)$$

Onde:

$P_w$  = densidade da potência do vento ( $W/m^2$ );

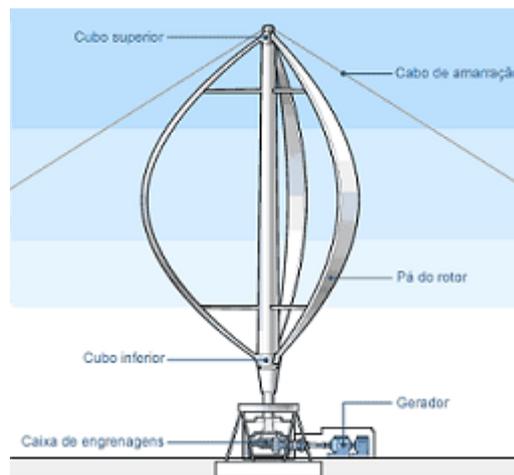
$\rho$  = densidade do ar ( $Kg/m^3$ );

$V$  = velocidade do vento ( $m/s$ ).

A equação exibida retrata o quão importante é a velocidade do vento para a geração de eletricidade. Em conformidade com Li et al. (2001), a densidade do ar pouco influencia, uma vez que  $\rho$  possui um pequeno intervalo de variações, ao passo que, a velocidade do vento está sujeita a variações acentuadas e, para qualquer flutuação, é elevada à terceira potência.

As turbinas eólicas podem possuir eixo vertical ou horizontal.

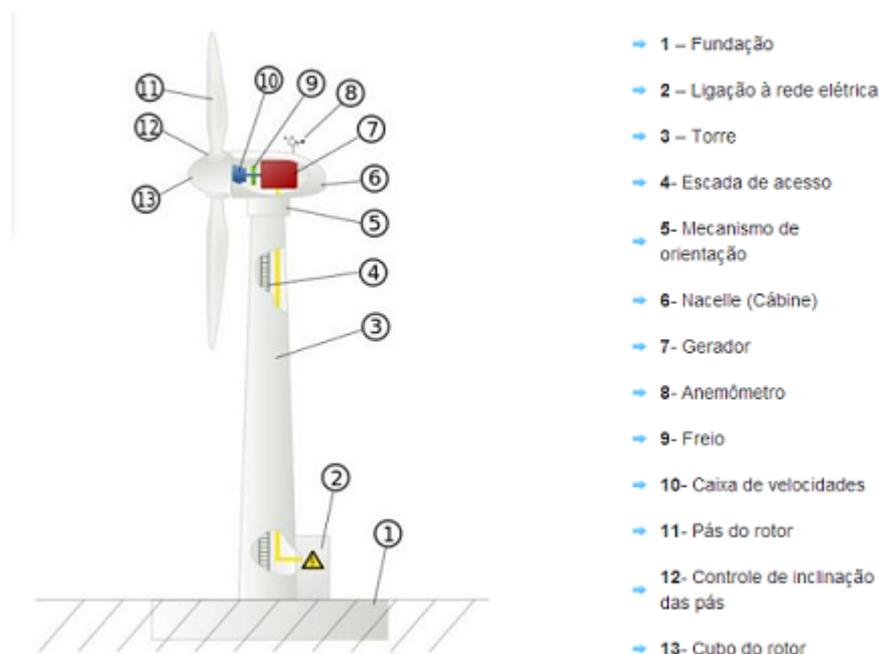
Turbinas de eixo vertical são pouco comuns. Segundo Rossi e Oliveira (2002), esse tipo de turbina possui as vantagens de não necessitar de ajustes no rotor quando o vento muda de direção e de os serviços de instalação e manutenção serem facilitados pelo fato de os equipamentos se localizarem no nível do solo. Por outro lado, estes aeroogeradores precisam de um impulso de seu sistema elétrico para dar partida. Ademais, a pouca elevação do rotor diminui o desempenho do aeroogerador, já que muitas vezes a topografia reduz a velocidade do vento. A Figura 2 exhibe a estrutura de uma turbina de eixo vertical:



**Figura 2: Turbina de eixo vertical.**

[Fonte: <http://ambiente.hsw.uol.com.br/energia-eolica1.htm>]

Os aerogeradores de eixo horizontal, apesar de necessitarem de uma maior velocidade do vento para rotacionar, são mais eficientes em situações em que as condições do vento são boas. A Figura 3, a seguir, apresenta os componentes de uma turbina de eixo horizontal:



**Figura 3: Turbina de eixo horizontal.**

[Fonte: <http://www.publicarartigos.com/componentes-aerogerador.php>]

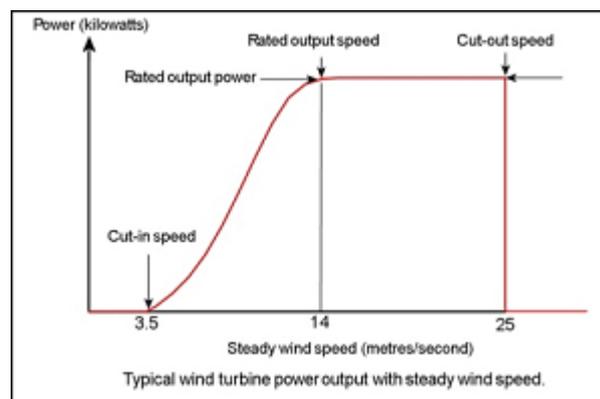
Para finalizar a abordagem sobre aerogeradores, é relevante expor os seus impactos na natureza. Os principais impactos são:

- ruído - turbinas de grande porte geram um ruído expressivo, em razão disto existe regulamentação própria para a instalação de aerogeradores próximos a áreas residenciais;
- mortalidade de aves - pássaros migratórios colidem com as hélices;
- poluição visual - os aerogeradores são bastante visíveis, o que interfere nas paisagens naturais, podendo haver restrições quanto à instalação em regiões turísticas de esplendor natural.

Nota-se, portanto, que são consequências de baixas proporções quando comparadas a outras matrizes energéticas.

### 2.1.3 Curva de Potência

De acordo com Wan, Ela e Orwig (2010), a curva de potência de um aerogerador consiste na representação gráfica da potência elétrica gerada em função da velocidade do vento. A partir de uma curva de potência, a produção de energia pode ser prevista sem conhecimentos específicos sobre a turbina e seus componentes. A Figura 4 exhibe um exemplo de curva de potência:



**Figura 4: Curva de potência.**

[Fonte: <http://www.wind-power-program.com>]

Ao analisar a curva de potência, percebe-se que é necessária uma velocidade mínima,

denominada *cut-in*, para que o vento rotacione as pás da turbina e inicie a produção de energia elétrica.

A partir da velocidade de *cut-in*, a potência gerada cresce de forma proporcional ao aumento da velocidade do vento. O limite máximo que um gerador pode produzir de energia é chamado de *rated output power*, e a velocidade do vento na qual se alcança essa produção é chamada de *rated output speed*.

Quando a *rated output speed* é excedida, as lâminas do aerogerador são ajustadas para manter a produção de energia constante, não ultrapassando a quantidade máxima para a qual a turbina foi projetada. Caso a velocidade do vento continue a aumentar, há risco de danificar o rotor. Portanto, a partir da velocidade de *cut-out*, é acionado um mecanismo de paralisação do rotor, interrompendo a produção de energia.

### 2.1.4 Potencial Brasileiro de Geração de Energia Eólica

O potencial elétrico total que a energia eólica pode originar no Brasil é de 143 mil MW. Esta elevada quantia corresponde a pouco mais do que 10 usinas de Itaipu. Além deste potencial, as jazidas de vento do país se beneficiam por não estarem sujeitas a ciclones e fortes turbulências como nos Estados Unidos e Europa. O potencial elétrico instalado no país é de 114 mil MW, levando-se em conta todas as matrizes utilizadas (JANNUZZI, 2012).

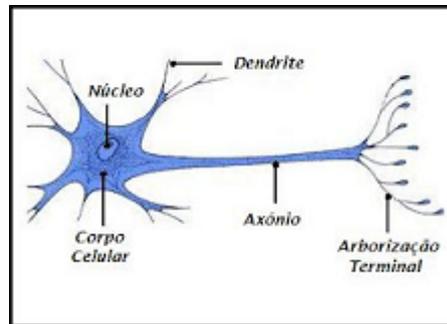
As regiões com maiores potenciais são a Nordeste, Sudeste e Sul, cada qual com 75 mil MW, 29,7 mil MW e 22,8 mil MW, respectivamente. Segundo Jannuzzi (2012), este potencial ainda é maior ao se considerar parques que podem ser instalados no mar (*off-shore*). Esse tipo de instalação possui como principal vantagem o fato de o mar apresentar uma superfície de baixa rugosidade, o que possibilita que o vento circule facilmente, sem barreiras. Entretanto, há um maior custo de instalação e manutenção.

## 2.2 Redes Neurais Artificiais

O cérebro humano é um processador poderoso de informações. Amparado em suas unidades funcionais, os neurônios, o cérebro trabalha de forma complexa, não-linear e paralela. Esse potente órgão é composto por cerca de 10 bilhões de neurônios, os quais estão conectados entre si formando uma intensa rede capaz de realizar aproximadamente 60 trilhões de ligações sinápticas (Valença, 2010)

O neurônio é a célula do sistema nervoso. Ele é responsável pela propagação do

impulso nervoso, mensagem que conduz informação do sistema nervoso para todo o organismo. Essa célula nervosa possui a estrutura, simplificada, exibida na Figura 5:



**Figura 5: Neurônio biológico.**

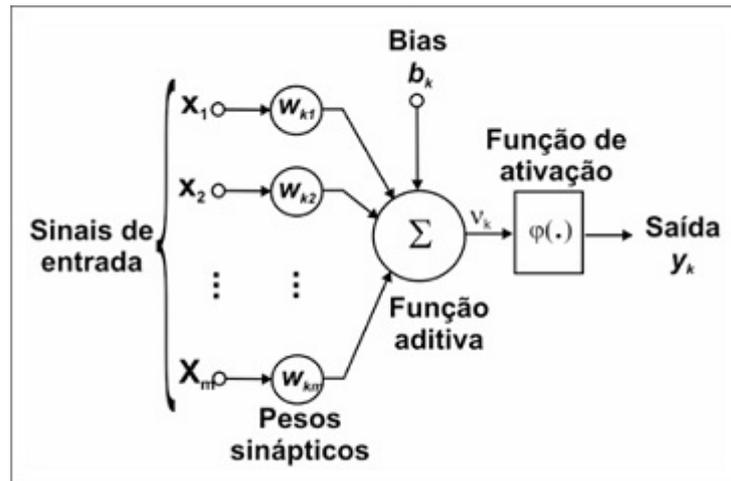
[Fonte: <http://psico12paratodos.blogspot.com.br/2013/05/neuronio.html>]

Os dendritos ou dendrites são ramificações cuja função é a recepção de estímulos provenientes do ambiente ou de outros neurônios, conduzindo-os para o corpo celular. O corpo celular ou soma é onde ocorre os processos metabólicos. O axônio, por sua vez, conduz o estímulo oriundo do corpo celular para outras células do sistema. A arborização terminal é a parte final do axônio e também é denominada telodendro.

A transmissão do impulso nervoso entre dois neurônios ocorre na região de sinapse. A comunicação é realizada do axônio do neurônio transmissor para o dendrito do receptor.

Inspiradas no funcionamento do cérebro, as RNAs constituem um ramo da Inteligência Artificial e podem ser definidas como modelos matemáticos cujo processo de aprendizagem ocorre por meio da experiência.

A primeira proposição de um neurônio artificial foi elaborada por McCulloch e Pitts, em 1943. De acordo com Kovács (2006), o neurônio criado era demasiadamente simples (Figura 6).



**Figura 6: Neurônio proposto por McCulloch e Pitts.**

[Fonte: [http://www.eps.ufsc.br/teses/todesco/capit\\_3/imagens/f33\\_tod.htm](http://www.eps.ufsc.br/teses/todesco/capit_3/imagens/f33_tod.htm)]

A sua saída pode ser pulso ou não pulso. Para calcular se o neurônio será ativado ou não, calcula-se a entrada líquida a partir da seguinte soma ponderada:

$$net_i = \sum_{j=1}^n w_{ij}x_j - \theta \quad (2.2)$$

Onde:

$net_i$  = entrada líquida;

$w_{ij}$  = pesos sinápticos;

$x_j$  = variáveis de entrada;

$\theta$  = limiar.

Se a entrada líquida for maior ou igual a zero, o pulso é emitido (a saída é 1), caso contrário não há propagação (a saída é 0).

Posteriormente, Frank Rosenblatt (1958), com base no neurônio de McCulloch e Pitts, concebeu o mais simples modelo de RNA: o *Perceptron*. Segundo Valença (2010), o *Perceptron* pode ser definido como uma rede neural constituída por uma camada de entrada onde estão as informações que alimentam a rede, uma função soma que pondera as entradas através dos pesos sinápticos e uma função de ativação (função degrau) que determinará qual o sinal emitido pelos neurônios da camada de saída.

O *Perceptron*, entretanto, é limitado pelo fato de apenas lidar com saídas discretas. Em razão disto, Widrow e Hoff desenvolveram a ADALINE, cuja diferença para o *Perceptron* reside na possibilidade de utilização de uma função contínua. As mais comumente

empregadas são: a função linear, a função sigmóide logística e a função tangente hiperbólica.

Com base nas redes mencionadas, surgiu uma importante classe de redes neurais: *Perceptron* de múltiplas camadas (do inglês, *Multilayer Perceptron*, MLP).

## 2.3 Redes MLP

A rede MLP é considerada uma generalização do *Perceptron*. A adição de pelo menos uma camada intermediária (camada escondida) possibilita a resolução de problemas não-linearmente separáveis. Isto faz da MLP uma ferramenta poderosa, uma aproximadora universal de funções.

Uma rede MLP apresenta três tipos de camadas. A de entrada é composta por neurônios que representam as variáveis de entrada do problema. A camada intermediária (pode haver mais de uma) é a responsável pela não-linearidade da rede, seus neurônios possuem, geralmente, uma função sigmóide de ativação, podendo ser a logística ou a tangente hiperbólica (Valença, 2010). Por último, a camada de saída, responsável pela resposta da rede, representa a variável que se pretende classificar ou prever. Seus neurônios também possuem função de ativação (linear, sigmóide logística ou tangente hiperbólica).

### 2.3.1 Treinamento com algoritmo *backpropagation*

O treinamento de uma rede MLP é supervisionado, ou seja, para cada entrada existente no conjunto de treinamento, sabe-se qual deve ser a saída. A partir do resultado esperado, ajustam-se os pesos das ligações de forma que a saída da rede se aproxime da desejada. Entretanto, na MLP há a dificuldade de calcular os novos pesos da camada intermediária, visto que não se sabe qual deveria ser a saída desta camada. Para solucionar este problema, Paul Werbos desenvolveu o algoritmo *backpropagation*, o qual realiza a propagação recursiva dos erros.

De acordo com Valença (2010), esse algoritmo consiste basicamente em dois passos. No primeiro, o sinal se propaga da entrada da rede até a saída, onde é calculado o erro. No segundo passo, os erros são propagados recursivamente da camada de saída até a de entrada, é nesta etapa que os pesos são ajustados. O ajuste dos pesos é realizado da seguinte forma:

$$w_{i,j}^m(novo) = w_{i,j}^m(antigo) + \alpha \delta_i^m f^{m-1}(net_j^{m-1}) \quad (2.3)$$

Onde:  $w_{i,j}^m$  é o peso sináptico entre o neurônio emissor  $j$  da camada  $m$  e o neurônio receptor  $i$ ;  $\alpha$  é a taxa de aprendizagem;  $f^{m-1}$  são os sinais de entrada emitidos pelos neurônios da camada anterior e  $\delta_i^m$  é conhecido como sensibilidade.

A sensibilidade pode ser calculada de acordo com a equação 2.4:

$$\delta_j^{m-1} = f'^{m-1}(net_j^{m-1}) \sum_{i=1}^{Nneuronios} w_{ij}^m \delta_i^m \quad (2.4)$$

Nesta equação, percebe-se que a sensibilidade é calculada da última para a primeira camada. O termo  $f'^{m-1}(net_j^{m-1})$  é a derivada da função de ativação dos neurônios da camada  $m - 1$  em relação à entrada líquida. O valor  $Nneuronios$  é a quantidade de neurônios que existem na camada que recebe o sinal.

No cálculo da sensibilidade dos neurônios da camada de saída, a seguinte expressão é utilizada:

$$\delta_i^m = (d_i - y_i) f'(net_i) \quad (2.5)$$

Onde  $d_i$  é a saída esperada,  $y_i$  é a saída calculada e  $f'(net_i)$  é a derivada da função de ativação dos neurônios da camada de saída.

### 2.3.2 Treinamento com algoritmo de Levenberg-Marquardt

O algoritmo de treinamento *backpropagation* tende a convergir muito lentamente, acarretando num elevado custo computacional (SILVA; SPATTI; FLAUZINO, 2010). Em razão desta desvantagem, foram propostos algoritmos de treinamento com o objetivo de tornar o processo de convergência mais eficiente. Entre tais aperfeiçoamentos, há o algoritmo de treinamento de Levenberg-Marquardt, utilizado nos experimentos deste trabalho.

No método Levenberg-Marquardt, a atualização dos pesos ocorre segundo a equação a seguir:

$$\Delta W = (J^T(W) \cdot J(W) + \mu \cdot I)^{-1} \cdot J^T(W) \cdot E \quad (2.6)$$

Onde:  $\Delta W$  representa o ajuste do peso;  $J(W)$  é a matriz Jacobiana, matriz de derivadas de primeira ordem do erro em relação ao peso;  $J^T(W)$  é a matriz Jacobiana transposta;  $I$  é a matriz identidade;  $\mu$  é uma constante que ajusta a taxa de convergência do algoritmo e  $E$  é o vetor de erro.

De acordo com Silva, Spatti e Flauzino (2010), ao implementar o método desenvolvido por Levenberg e Marquardt, é comprovado que o treinamento de redes MLP apresenta um desempenho na ordem de 10 a 100 vezes mais rápido do que o algoritmo *backpropagation*

tradicional.

## 2.4 Redes com Função de Base Radial

Outra classe de RNA utilizada neste trabalho foi a Rede com Função de Base Radial (do inglês, *Radial Basis Function*, RBF). Esta RNA, tal qual a MLP, é qualificada como aproximadora universal de funções, apresentando-se como uma poderosa ferramenta largamente utilizada em problemas que envolvem previsões e classificação de padrões.

A RBF, na sua constituição típica, é composta por três camadas: a de entrada, a escondida e a de saída. O principal fator que diferencia esse tipo de rede das redes MLP é a função de ativação dos neurônios da camada escondida, que deve ser uma função de base radial. Uma função de base radial é aquela cuja saída depende da distância entre o valor de entrada e um determinado centro, pode-se citar como exemplo a função de Gauss (equação 2.7):

$$g(u) = e^{-\frac{(u-c)^2}{2\sigma^2}} \quad (2.7)$$

Onde  $u$  é a entrada,  $c$  é o centro,  $\sigma$  denota o desvio padrão.

A camada de saída, por sua vez, é composta por neurônios cuja ativação se dá por meio de funções lineares.

### 2.4.1 Treinamento de redes RBF

O processo de aprendizagem de uma rede RBF é composto por duas etapas. Na primeira, são ajustados os pesos dos neurônios da camada intermediária por meio da aplicação de um método de aprendizagem não-supervisionado, isto é, a rede irá adquirir conhecimento a partir dos dados de entrada apenas, não sendo necessária a apresentação de suas respectivas saídas. Já na segunda etapa, realiza-se os ajustes dos neurônios da camada de saída, empregando um método de aprendizado supervisionado.

A partir do exposto, pode-se concluir que, diferentemente do que ocorre numa rede MLP, o procedimento de treinamento de redes RBF inicia-se nos neurônios da camada oculta e encerra-se nos da camada de saída.

A primeira etapa do treinamento se destina ao ajuste dos parâmetros da função de base radial empregada. Tais parâmetros são: o centro da função e a variância.

De acordo com Silva, Spatti e Flauzino (2010), o centro da função de ativação de um neurônio da camada escondida está diretamente associado aos seus próprios pesos. Por exemplo, se a função de Gauss for escolhida como função de ativação, a saída de cada

neurônio  $j$  da camada intermediária será:

$$g_j^{(1)}(x) = e^{-\frac{\sum_{i=1}^n (x_i - W_{ji}^{(1)})^2}{2\sigma_j^2}} \quad (2.8)$$

Onde:  $n$  é a quantidade de entradas da rede;  $x_i$  é o valor da  $i$ -ésima entrada;  $W_{ji}$  representa o valor do peso sináptico que conecta a  $i$ -ésima entrada ao  $j$ -ésimo neurônio da camada oculta e  $\sigma_j^2$  é a variância associada ao  $j$ -ésimo neurônio da camada oculta.

Os centros podem ser determinados a partir da utilização de algoritmos de *clusterização*. Um método bastante utilizado é o  $k$ -médias, cujo desígnio é posicionar os centros de  $k$ -gaussianas em regiões em que os padrões de entrada tenderão a se agrupar. É importante esclarecer que o valor de  $k$  corresponde à quantidade de neurônios da camada intermediária, já que a função de ativação de cada um é uma gaussiana e seus centros serão representados pelos seus respectivos vetores de peso.

Durante a execução do algoritmo  $k$ -médias, são calculadas as distâncias euclidianas entre o valor de cada exemplo de entrada e o respectivo peso que o conecta ao neurônio da camada oculta. Cada exemplo é atribuído ao grupo do neurônio mais próximo. Posteriormente, os pesos são ajustados conforme a seguinte equação:

$$W_{ji} = \frac{1}{m^{(j)}} \sum_{x^{(k)} \in \Omega^{(j)}} x^{(k)} \quad (2.9)$$

Em que:  $W_{ji}$  é o valor do peso da ligação entre a  $i$ -ésima entrada e o  $j$ -ésimo neurônio;  $\Omega^{(j)}$  é o grupo de amostras atribuídas ao neurônio  $j$ ;  $m^{(j)}$  é o número de amostras em  $\Omega^{(j)}$  e  $x^{(k)}$  é o valor da  $k$ -ésima amostra. Este procedimento, desde a execução do algoritmo  $k$ -médias até o ajuste dos pesos se repete até que não haja mudanças nos grupos  $\Omega^{(j)}$ .

Com os pesos já ajustados, calcula-se a variância de cada uma das funções de ativação aplicando-se a seguinte equação:

$$\sigma_j^2 = \frac{1}{m^{(j)}} \sum_{x^{(k)} \in \Omega^{(j)}} \sum_{i=1}^n (x_i^{(k)} - W_{ji})^2 \quad (2.10)$$

Em que  $\Omega^{(j)}$  é o grupo de amostras atribuídas ao neurônio  $j$ ,  $m^j$  é a quantidade de amostras em  $\Omega^{(j)}$ ,  $n$  é o número de entradas da rede,  $x_i^{(k)}$  é o valor que a entrada  $i$  assume quando a  $k$ -ésima amostra é apresentada e  $W_{ji}$  é o valor do peso da ligação entre a  $i$ -ésima entrada e o  $j$ -ésimo neurônio.

Ao finalizar a etapa do treinamento não-supervisionado e de posse dos parâmetros das funções de base radial, realiza-se a segunda fase do treinamento. Neste segundo estágio, utiliza-se os mesmos procedimentos empregados para a camada de saída de uma rede

MLP.

O conjunto de treinamento será, então, constituído por pares de entrada e saída desejada, em que as entradas serão as respostas geradas pelas funções de transferência gaussianas dos neurônios que compõem a camada escondida associadas aos pesos e limiares referentes ao neurônio (SILVA; SPATTI; FLAUZINO, 2010).

Após a execução das duas etapas do treinamento, a rede RBF está apta para estimar as saídas quando da apresentação de novas amostras de entradas para o processo que foi mapeado.

## ***3 Metodologia***

Este capítulo aborda as estratégias e implementações desenvolvidas com base na fundamentação teórica apresentada para perseguir os objetivos deste trabalho.

### **3.1 Base de Dados**

A base de dados a partir da qual os experimentos foram elaborados é uma base de dados de um parque eólico real que será chamado aqui de parque A. Os dados utilizados abrangem o período de 1<sup>o</sup> de dezembro de 2011 a 31 de julho de 2012, com intervalo de 30 minutos. Estes dados são: a velocidade média do vento, sua direção média e a potência média gerada.

Entre os dados concedidos existem períodos em que não houve coleta de valores em razão de falhas ocorridas no momento da medição dos valores. Tais períodos falhos foram retirados da base de dados. Não há a expectativa que esta ação cause grandes impactos nos resultados finais, visto que é grande a quantidade de dados disponíveis.

### **3.2 Pré-processamento dos Dados**

Antes que os dados fossem apresentados como entradas, eles foram submetidos ao processo de normalização. De acordo com Valença (2010), “todas as variáveis devem ser normalizadas para assegurar que elas recebam igual atenção durante o processo de treinamento”.

A normalização garante que os valores dos dados inseridos na rede neural estejam contidos num mesmo intervalo numérico. Isto evita que variáveis que assumem valores altos se sobreponham às que apresentam valores relativamente baixos.

Outra grande importância da normalização reside no fato de que as redes neurais não têm a capacidade de generalizar quando utilizadas em um conjunto de entradas que apresente um intervalo de variação distinto ao do treinamento.

O tipo de normalização aplicada foi a transformação linear, obedecendo à Equação 3.1:

$$y = \frac{(b - a)(x_i - x_{min})}{x_{max} - x_{min}} + a \quad (3.1)$$

Onde:  $y$  é o valor normalizado;  $x_i$  é o valor original;  $x_{min}$  assume o valor mínimo de  $x$ ;  $x_{max}$  é o valor máximo de  $x$ ;  $a$  e  $b$  são os limites escolhidos sendo, neste trabalho, atribuído -1 a  $a$  e 1 a  $b$ .

É importante informar que não só as entradas foram submetidas ao procedimento de normalização, mas os valores das saídas desejadas também foram limitadas entre [-1 e 1]. Portanto, quando a rede produz uma saída para uma determinada entrada, realiza-se uma transformação inversa para a obtenção de seu valor na escala original.

### 3.3 Construção da Curva de Potência Utilizando Polinômio de 2º Grau

Os fabricantes de turbinas eólicas utilizam funções polinomiais de grau 2 para a construção de curvas de potência teóricas. Entretanto, em situações reais, muitas vezes a potência gerada não condiz precisamente com a estimada já que as condições atmosféricas em que a turbina foi testada divergem daquelas em que ela é utilizada de fato.

Nos experimentos realizados, 70% dos dados foram utilizados para treinamento, isto é, foram empregados no cálculo dos coeficientes do polinômio. Os coeficientes computados são os que melhor ajustam a curva polinomial aos dados, de forma a reduzir o erro entre a saída do polinômio e a saída desejada. De posse dos coeficientes, os 30% restantes dos dados foram destinados aos testes de desempenho.

#### 3.3.1 Características da Implementação

Na realização dos experimentos envolvendo polinômio de 2º, foi utilizada a função **polyfit** do MATLAB, descrita a seguir:

$$p = \text{polyfit}(x, y, n) \quad (3.2)$$

Onde:  $p$  é um vetor que contém os coeficientes calculados;  $x$  é o vetor de entradas (velocidades do vento);  $y$  é o vetor de saídas desejadas (potências elétricas reais) e  $n$  é o grau do polinômio.

Após calcular os coeficientes, utiliza-se a função **polyval** (Equação 3.3) do MATLAB

para obter a saída proveniente do polinômio.

$$y = \text{polyval}(p, x) \quad (3.3)$$

Na Equação 3.3,  $y$  é o vetor de saídas (potências previstas) calculadas com o emprego do polinômio;  $p$  é o vetor que contém os coeficientes polinomiais calculados anteriormente e  $x$  representa o vetor de entradas (velocidades do vento).

### 3.4 Construção da Curva de Potência Utilizando Função Sigmóide

Outra função utilizada pelos fabricantes de aerogeradores para a construção da curvas de potências teóricas é a sigmóide.

A função sigmóide escolhida para este trabalho está descrita na Equação 3.4:

$$f(x) = \frac{a_0 + a_1}{1 + e^{\left(-\frac{x-a_2}{a_3}\right)}} \quad (3.4)$$

Onde  $a_0$ ,  $a_1$ ,  $a_2$  e  $a_3$  são os coeficientes da função.

70% da base de dados foi destinado ao cálculo dos coeficientes que melhor adequam a curva sigmóide aos dados. Num segundo momento, 30% da base foi empregado na realização dos testes de desempenho da função.

#### 3.4.1 Características da Implementação

No cenário da implementação no MATLAB, a função sigmóide foi definida como a seguinte função:

$$f = @(p, x) p(1) + p(2) ./ (1 + \exp(-(x - p(3))/p(4))) \quad (3.5)$$

Onde  $p(n)$  representa os coeficientes da função e  $x$  é a variável de entrada.

Após definir a função sigmóide a ser empregada, utiliza-se a função **nlinfit** da Equação 3.6 para calcular os coeficientes que melhor ajustam a curva sigmoidal aos dados.

$$p = \text{nlinfit}(X, Y, f, \text{coef0}) \quad (3.6)$$

Na Equação 3.6,  $p$  é o vetor de coeficientes que otimizam a função  $f$  para que, dado o vetor de entradas  $X$  (velocidades do vento), suas saídas se aproximem das desejadas  $Y$  (potências elétricas reais). Os valores iniciais dos coeficientes são especificados pelo parâ-

metro  $coef0$ . Neste trabalho, a partir de pesquisas e de procedimentos empíricos,  $coef0$  assumiu o seguinte valor: [0 20 50 5]. Esses coeficientes são conhecidos por construírem e moldarem uma curva sigmoïdal com formato bem definido.

Após o cálculo dos coeficientes, a função está pronta, e para ser utilizada basta chamá-la da seguinte forma:

$$y = f(p, X) \quad (3.7)$$

Onde:  $y$  é o vetor de saídas (potências previstas);  $f$  é a função previamente definida;  $p$  é o vetor de coeficientes calculados e  $X$  representa o vetor de entradas (velocidades do vento).

## 3.5 Construção da Curva de Potência Utilizando Redes MLP

Os experimentos envolvendo redes MLP foram realizados a partir de diferentes configurações de seus parâmetros. A seguir são explanados os procedimentos de modelagem das redes MLP empregadas nos experimentos.

### 3.5.1 Subdivisão dos Dados

Os dados foram divididos em três subconjuntos: 50% dos dados constituem o conjunto para o treinamento da rede, destinado ao ajuste dos pesos sinápticos da rede neural; 20% foram destinados à realização da validação cruzada, técnica utilizada para “ informar quando se deve parar de treinar a rede neural de forma a evitar que a rede perca sua capacidade de generalização ” (Valença, 2010); os 30% restantes foram reservados para a avaliação de desempenho do modelo quando concluída a fase de treinamento.

### 3.5.2 Arquitetura das Redes

Todas as redes MLP utilizadas apresentam 3 camadas (de entrada, intermediária e de saída). A função de ativação escolhida para os neurônios da camada escondida foi a tangente hiperbólica, representada na Equação 3.8:

$$y = f(net_i) = \frac{e^{net_i} - e^{-net_i}}{e^{net_i} + e^{-net_i}} \quad (3.8)$$

Onde  $net_i$  assume o valor da entrada líquida do neurônio  $i$ . Já a função de transferência escolhida para os neurônios da camada de saída foi a seguinte função linear:

$$y = f(net_i) = net_i \quad (3.9)$$

Onde  $net_i$  é a entrada líquida do neurônio.

Na camada de entrada, foram realizados experimentos variando a quantidade de neurônios, uma vez que cada neurônio representa uma variável considerada como entrada para o problema. Foram testadas combinações de entradas como: a velocidade do vento; a velocidade do vento em instantes passados; a direção do vento; a direção do vento em momentos anteriores.

Foi testado também um tratamento para a entrada da direção do vento no qual foram criados dois neurônios de entrada para representar a informação da direção do vento. O valor de cada um é calculado a partir das seguintes equações:

$$x_1 = \text{sen}\left(2\pi \frac{N}{180}\right) \quad (3.10)$$

$$x_2 = \text{cos}\left(2\pi \frac{N}{180}\right) \quad (3.11)$$

Onde  $N$  representa o valor da direção do vento. De acordo com Valença (2010), este tratamento é empregado em variáveis de natureza circular. Alternativamente, poderiam ter sido aplicadas outras operações em que uma variável representaria o complemento da outra, como tangente e cotangente.

A quantidade de neurônios presentes na camada intermediária variou ao longo dos experimentos. O número de neurônios na camada de saída foi 1, representando a potência energética prevista.

### 3.5.3 Treinamento

Na etapa de treinamento da rede, a cada ciclo, as amostras de entradas foram apresentadas numa ordem aleatória. Isto confere uma maior estocasticidade ao processo, evitando que a rede memorize os padrões a partir da ordem em que são apresentados.

Ainda na fase de treinamento, adotou-se a atualização em lote (*batch*) dos pesos, isto é, os ajustes apenas são realizados após a apresentação de todos os exemplos de treinamento que compõem uma época. Também chamada de atualização *off-line*, ela computa o ajuste dos pesos para cada amostra e, ao fim do ciclo, soma os ajustes de cada ligação

sináptica e calcula a média do  $\Delta W_{ij}$  para então efetivar a correção.

Foram realizados experimentos tanto com o algoritmo *backpropagation* quanto com o de Levenberg-Marquardt. No caso do *backpropagation*, a taxa de aprendizado foi igual a 0,8.

### 3.5.3.1 Critério de Parada

O mecanismo adotado para a parada do treinamento da rede foi a validação cruzada. Ao fim de cada ciclo ou época, que é a apresentação de todos os exemplos de entrada, calcula-se a diferença entre a saída obtida da rede e o valor desejado. Este erro é calculado tanto para o conjunto de treinamento empregado para ajustar os pesos (neste trabalho foram utilizados 50% dos dados) quanto para o conjunto de validação cruzada (foram utilizados 20% dos dados neste trabalho). A cada ciclo, os erros para estes dois conjuntos de dados são avaliados. Enquanto o erro da validação cruzada continuar diminuindo, a rede está aprendendo corretamente. Porém, quando o erro da validação cruzada aumentar enquanto o erro do conjunto de treinamento continuar a decrescer, é sinal que a rede está ficando superajustada (*overfitting*) e é o momento de parar o treinamento para que a rede não memorize os padrões de treinamento.

Neste trabalho, adotou-se que o treinamento irá parar quando o erro da validação cruzada aumentar durante 6 épocas sucessivas. É importante destacar que a validação cruzada também minimiza o risco de subajustamento (*underfitting*) da rede situação em que o treinamento ocorre em menos ciclos do que o necessário, implicando uma baixa capacidade de generalização.

## 3.5.4 Características da Implementação

Em relação à implementação no MATLAB, utilizou-se a função **feedforwardnet** para criar uma rede MLP, descrita a seguir:

$$net0 = feedforwardnet(n) \quad (3.12)$$

Onde *net0* representa a rede criada e *n* é a quantidade de neurônios na camada intermediária.

Posteriormente, utiliza-se a função **configure** (3.13) para ajustar a arquitetura da rede de acordo com a quantidade de variáveis de entrada e de saída. É por meio dessa

função, também, que os pesos das ligações sinápticas são inicializados.

$$net = configure(net0, x, t) \quad (3.13)$$

Na Equação 3.13, *net* representa a rede neural configurada, *net0* é a rede previamente criada cuja quantidade de neurônios na camada oculta já foi definida, *x* é a matriz que contém os dados de entrada a partir da qual são criados os neurônios da camada de entrada e *t* é a matriz constituída pelas saídas desejadas da rede e é a partir dela que são criados os neurônios da camada de saída.

Em seguida, são definidas as porcentagens segundo as quais os dados serão subdivididos. A seguir, é exibido como foram definidas as proporções da divisão dos dados:

$$net.divideParam.trainRatio = 0.5 \quad (3.14)$$

$$net.divideParam.valRatio = 0.2 \quad (3.15)$$

$$net.divideParam.testRatio = 0.3 \quad (3.16)$$

As Equações 3.14, 3.15 e 3.15 definem os tamanhos percentuais dos conjuntos de treinamento, validação e de teste, respectivamente.

Para escolher o algoritmo de treinamento, Levenberg-Marquardt, utilizou-se o seguinte comando:

$$net.trainFcn = 'trainlm' \quad (3.17)$$

Na situação em que o algoritmo de treinamento foi o *backpropagation*, substituiu-se o “trainlm” por “trainrp” e configurou-se a taxa de aprendizado do seguinte modo:

$$net.trainParam.lr = 0.8 \quad (3.18)$$

O intervalo para o qual os dados foram normalizados foi definido da seguinte maneira:

$$net.inputs\{1\}.processParams\{2\}.ymin = -1 \quad (3.19)$$

$$net.inputs\{1\}.processParams\{2\}.ymax = 1 \quad (3.20)$$

As Equações 3.19 e 3.20 definem os limites inferior e superior do intervalo de normalização, respectivamente.

A função de transferência dos neurônios da camada escondida, a tangente hiperbólica, foi definida da maneira a seguir:

$$net.layers\{1\}.transferFcn = 'tansig' \quad (3.21)$$

A Equação 3.22 define a função de ativação dos neurônios da camada de saída.

$$net.layers\{2\}.transferFcn = 'purelin' \quad (3.22)$$

O treinamento da rede é realizado por meio da função **train**, descrita a seguir:

$$trainedNet = train(net, x, t) \quad (3.23)$$

Onde: *trainedNet* é a rede com os pesos sinápticos já ajustados pelo treinamento; *net* é a rede que irá passar pelo processo de treinamento; *x* é a matriz formada pelos dados de entrada e *t* representa a matriz composta pelos dados de saída.

Por fim, a saída calculada pela rede é obtida da seguinte forma:

$$y = trainedNet(x) \quad (3.24)$$

Onde *y* é saída calculada e *x* é a matriz de entradas.

## 3.6 Construção da Curva de Potência Utilizando Redes RBF

Por fim, nos últimos experimentos realizados, a curva de potência foi construída com o auxílio de redes RBF.

### 3.6.1 Subdivisão dos Dados

A base de dados foi dividida em dois subconjuntos. 70% dos dados constituem o agrupamento destinado ao treinamento da rede, a fim de procurar os pesos ótimos para o problema. Os 30% restantes, por sua vez, foram utilizados nos testes de desempenho da rede. Para a rede RBF, no MATLAB, não há o conjunto de validação cruzada já que é configurado previamente qual o erro médio quadrático que deve ser atingido. Mediante a realização de experimentos, verificou-se que, para o problema abordado neste trabalho, o treinamento da RBF deve parar quando o erro médio quadrático chegar ao valor de 16,00. Ao definir um erro médio quadrático inferior a 16,00, a rede ficou iterando repetidas vezes durante a etapa de treinamento sem conseguir alcançar o erro, isto é, a rede estagnou de forma a não melhorar sua precisão.

### 3.6.2 Arquitetura da Rede

A rede RBF empregada nos experimentos possui três camadas. Na camada de entrada, assim como nos experimentos com a rede MLP, a quantidade de neurônios variou a cada experimento, realizando-se combinações entre variáveis como: a velocidade do vento; a velocidade do vento em instantes anteriores; a direção do vento; a direção do vento em instantes anteriores. Já na camada intermediária, no início do treinamento da rede, há apenas um neurônio, a cada época adiciona-se um neurônio caso o erro médio entre a saída desejada e a obtida não seja inferior ao valor predefinido ou até que seja alcançado um número máximo de neurônios (definido como a quantidade de exemplos de entradas). Por fim, a camada de saída possui apenas um neurônio o qual representa a potência energética como saída da rede.

A função de transferência dos neurônios da camada oculta utilizada no MATLAB é:

$$y = f(n) = e^{-n^2} \quad (3.25)$$

Onde  $n$  é a entrada do neurônio calculada a partir do somatório das distâncias entre seus pesos sinápticos e as entradas da rede. O desvio padrão da função radial é definido antecipadamente, por experimentos cujo valor escolhido foi 1,0 ou 2,0.

A função de ativação da camada de saída escolhida foi a identidade.

Assim como na rede MLP, a cada ciclo do treinamento, os exemplos de entrada foram apresentados aleatoriamente. E as atualizações dos pesos foram realizadas em lotes (*batch*).

### 3.6.3 Características da Implementação

Para criar a rede RBF no MATLAB utilizou-se a função **newrb**, descrita a seguir:

$$net = newrb(P, T, GOAL, SPREAD) \quad (3.26)$$

Onde:  $net$  é a rede criada;  $P$  é a matriz com os dados de entrada;  $T$  representa a matriz com as saídas desejadas;  $GOAL$  é o erro mínimo quadrado almejado e  $SPREAD$  representa o desvio padrão escolhido para a função de base radial. Ao criar-se a rede, inicia-se automaticamente o processo de treinamento.

Finalizado o processo de treinamento, a rede está com os pesos ajustados e pronta para ser testada. As saídas provenientes da rede são obtidas da maneira a seguir:

$$y = net(X) \quad (3.27)$$

Onde *net* representa a rede criada e treinada e  $X$  é a matriz de entradas.

## 3.7 Execuções

Para possibilitar a análise estatística dos resultados, cada experimento foi executado 30 vezes. Nos primeiros experimentos, as execuções foram efetuadas mantendo-se fixo o conjunto de dados destinados aos testes. Posteriormente, percebeu-se que os dados que estavam sendo utilizados para a avaliação de desempenho poderiam não ser estatisticamente representativos da amostra populacional. Em virtude disso, foram realizados novos experimentos nos quais, a cada execução, os subconjuntos dos dados foram constituídos a partir de dados selecionados aleatoriamente.

Ao fim de cada execução, a partir da potência prevista, calcula-se o Erro Percentual Médio Absoluto (EPMA) em relação à potência desejada. O EPMA é calculado de acordo com a equação a seguir:

$$EPMA = \frac{\sum_{i=1}^n \left| \frac{d_i - y_i}{d_i} \right| 100}{n} \quad (3.28)$$

Na Equação 3.28,  $d_i$  é a saída desejada para a  $i$ -ésima previsão,  $y_i$  é a saída prevista para a  $i$ -ésima previsão e  $n$  representa a quantidade de previsões realizadas. Em seguida, somou-se os EPMA's das 30 execuções e foi calculada a média. A partir dos EPMA's, foi realizada a comparação entre os desempenhos dos métodos de previsão utilizados.

## 3.8 Testes Estatísticos

De posse dos resultados oriundos das 30 execuções de cada rede neural ou função, foram realizados testes estatísticos para certificar qual dos métodos obteve o melhor resultado na construção de curvas de potência.

O teste aplicado foi o da Soma dos Postos com Sinais de Wilcoxon para Pares Combinados, escolhido em virtude de as amostras não serem normalmente distribuídas. Para tanto, foi utilizado o *software* R, empregado em computações estatísticas.

O teste da Soma dos Postos com Sinais de Wilcoxon para Pares Combinados é um teste não-paramétrico (não faz suposições sobre a distribuição normal dos dados) que utiliza os postos de dados amostrais compostos de pares combinados. Tal teste objetiva averiguar a hipótese nula de que a população de diferenças tem mediana nula, o que significa que os dados são estatisticamente iguais. A hipótese alternativa, por sua vez, afirma que os pares de dados têm diferenças que vêm de uma população com mediana diferente

de zero.

O procedimento para testar a hipótese nula não é complexo. O passo-a-passo realizado é o seguinte:

1. calcula-se a diferença para cada par de dados selecionados aleatoriamente, mantém-se o sinal do resultado e descarta-se as diferenças iguais a zero;
2. ordena-se os resultados de forma crescente e os substitui pelos valores de suas posições (ou postos);
3. atribui-se a cada posto o sinal da diferença que o originou, calcula-se a soma dos postos positivos e a soma dos postos negativos;
4. atribui-se a  $T$  a menor das duas somas e a  $n$  o número de diferenças diferentes de zero;
5. se  $n < 30$  então a estatística de teste recebe o valor de  $T$ , caso contrário recebe o o valor proveniente de  $\frac{T - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}}$ ;
6. de posse dos valores de  $n$  e do nível de significância escolhido como 0,05, consulta-se a tabela de valores críticos do teste de Wilcoxon;
7. se a estatística de teste estiver na região crítica (se for menor do que ou igual ao valor crítico obtido da tabela) então a hipótese nula é rejeitada. Caso contrário, não se rejeita a hipótese nula.

O *software* R, por sua vez, ao computar a estatística de teste, calcula o *p-value* a ela associado. Caso o *p-value* seja menor que o nível de significância escolhido (0,05) então a hipótese nula é rejeitada. Caso contrário, aceita-se a hipótese nula.

## 4 *Resultados*

Este capítulo exhibe os resultados obtidos no presente trabalho a partir da aplicação da metodologia descrita no capítulo 3. Primeiramente, são apresentadas as curvas de potência elétrica construídas a partir de cada um dos métodos: polinômio do 2<sup>o</sup> grau; função sigmóide; diferentes arquiteturas de redes MLP e redes RBF. Depois, é apresentada a comparação entre eles.

### 4.1 Curva de Potência Construída com Polinômio do 2<sup>o</sup> Grau

O primeiro experimento utilizando o polinômio de grau 2, como mencionado anteriormente, foi realizado mantendo constante o conjunto de teste. O EPMA obtido é apresentado na tabela 1.

Posteriormente, reconheceu-se que os dados empregados no teste poderiam não ser representativos. Portanto, realizou-se um experimento em que a cada execução, de um total de 30, os dados fossem selecionados aleatoriamente para compor o conjunto de teste. A tabela 1 expõe os resultados encontrados.

**Tabela 1: Resultados dos Experimentos com Polinômio de 2<sup>o</sup> Grau**

[Fonte: elaboração própria]

experimento	modo de seleção do conj. de testes	EPMA
1	conjunto de testes constante	20.28
2	conjunto de testes aleatório	26.99

A curva de potência construída a partir do segundo experimento é representada pela Figura 7:

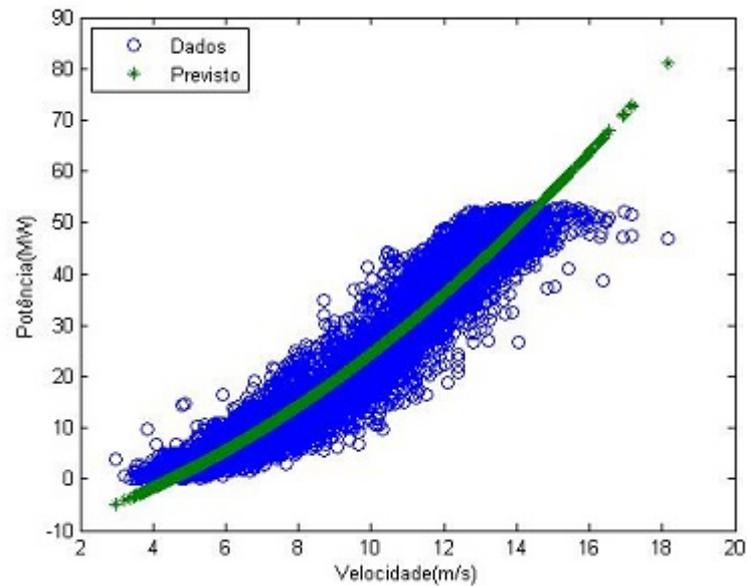


Figura 7: Curva de Potência Construída com Polinômio de 2º Grau.

[Fonte: elaboração própria]

## 4.2 Curva de Potência Construída com Função Sigmóide

Utilizando a função sigmoidal descrita no capítulo anterior, a tabela 2 apresenta os resultados obtidos:

Tabela 2: Resultados dos Experimentos com Função Sigmóide

[Fonte: elaboração própria]

experimento	modo de seleção do conj. de testes	EPMA
1	conjunto de testes constante	20.62
2	conjunto de testes aleatório	29.95

A diferença existente entre os resultados dos dois experimentos realizados com a função sigmóide evidencia que a utilização de um único conjunto de testes produziu resultados tendenciosos por não haver uma fiel representatividade entre os dados de testes e a população dos dados.

A curva de potência construída no segundo experimento com a função sigmóide é apresentada na Figura 8:

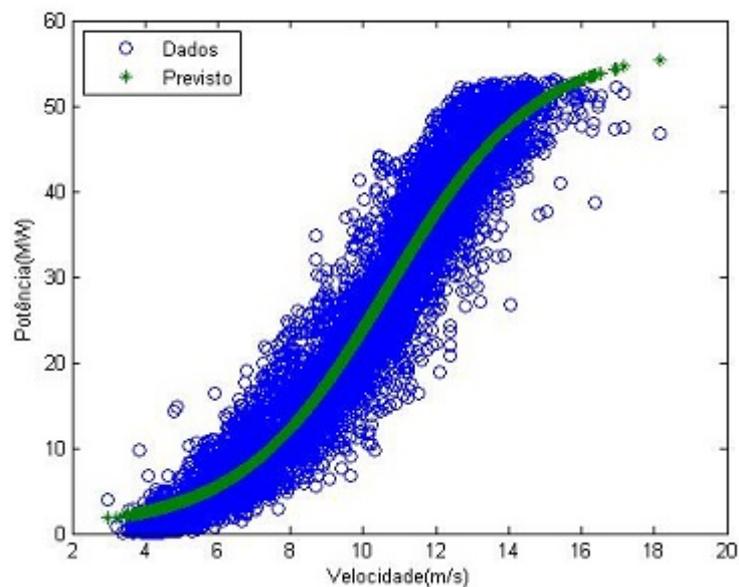


Figura 8: Curva de Potência Construída com Função Sigmóide.

[Fonte: elaboração própria]

### 4.3 Curva de Potência Construída com redes MLP

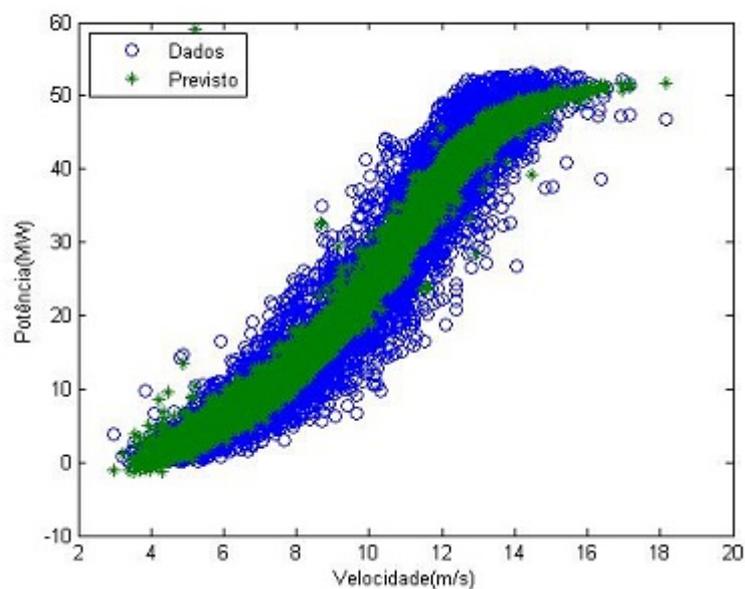
Conforme explicitado no capítulo anterior, foram realizados vários experimentos com redes MLP nos quais variou-se a maneira de selecionar os dados para o conjunto de teste, a entrada da rede, a quantidade de neurônios na camada escondida e o algoritmo de treinamento. Os resultados alcançados estão expressos na Tabela 3:

Tabela 3: Resultados dos Experimentos com Redes MLP

[Fonte: elaboração própria]

experimento	modo de seleção do conj. de testes	qtd. de neurônios na camada oculta	algoritmo de treinamento	entradas	EPMA
1	conjunto de testes constante	6	Levenberg-Marquardt	velocidades no tempo t	18.59
2	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades no tempo t	24.36
3	conjunto de testes aleatório	15	Levenberg-Marquardt	velocidades no tempo t	24.34
4	conjunto de testes aleatório	6	<i>Backpropagation</i>	velocidades no tempo t	27.03
5	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades no tempo t e t-30 min.	24.39
6	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades no tempo t, t-30, t-60, t-90, t-120, t-150 min.	24.93
7	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades e direções (sen e cos) no tempo t	24.75
8	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades e direções no tempo t, t-30, t-60, t-90 min.	24.23
9	conjunto de testes aleatório	6	Levenberg-Marquardt	velocidades em t até t-210 min. e direções no tempo t até t-90 min.	24.78

A Figura 9 exibe a curva de potência construída no experimento 8, o qual obteve o menor EPMA utilizando conjunto de testes constituído por dados selecionados de forma aleatória:



**Figura 9: Curva de Potência Construída com a Rede MLP do Experimento 8.**

[Fonte: elaboração própria]

Ao analisar a curva construída, percebe-se que a saída da rede se ajusta melhor aos dados do que quando utilizados os métodos anteriores. Em uma determinada velocidade do vento não há uma única potência prevista, pois há uma dependência em relação aos valores em instantes passados e às direções que o vento assume. Portanto, verifica-se que a curva construída é uma nuvem de pontos.

## 4.4 Curva de Potência Construída com redes RBF

A tabela 4 apresenta os resultados obtidos a partir da execução de redes RBF:

Tabela 4: Resultados dos Experimentos com Redes RBF

[Fonte: elaboração própria]

experimento	modo de seleção do conj. de testes	desvio padrão adotado	entradas	qtd. de neurônios na camada oculta ao fim da execução	EPMA
1	conjunto de testes constante	1	velocidades no tempo t	37	18.67
2	conjunto de testes aleatório	1	velocidades no tempo t	11	25.30
3	conjunto de testes aleatório	2	velocidades no tempo t	6	25.97
4	conjunto de testes aleatório	1	velocidades no tempo t e t-30 min.	39	29.48
5	conjunto de testes aleatório	2	velocidades no tempo t e t-30 min.	16	25.78
6	conjunto de testes aleatório	1	velocidades e direções (sen e cos) no tempo t	10	25,42
7	conjunto de testes aleatório	2	velocidades e direções (sen e cos) no tempo t	5	28.17

Por fim, a Figura 10 retrata a curva de potência proveniente do experimento 2 que obteve o menor EPMA a partir de dados de testes escolhidos aleatoriamente.

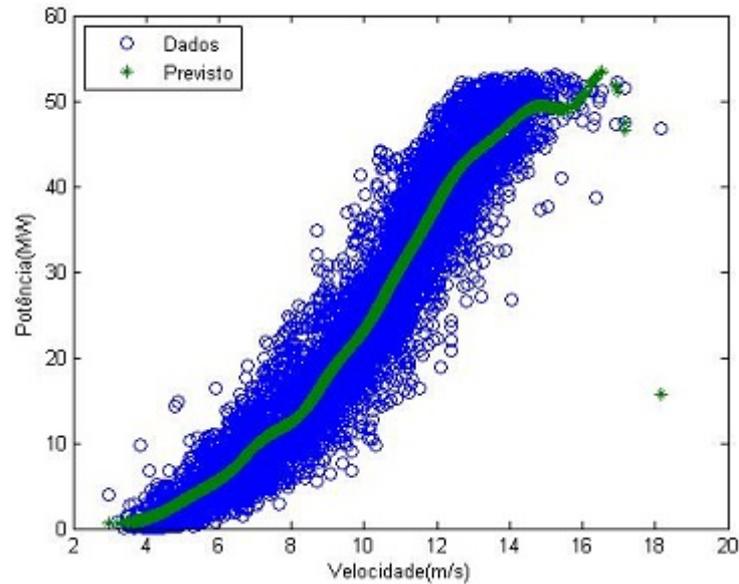


Figura 10: Curva de Potência Construída com a Rede RBF do Experimento 2.

[Fonte: elaboração própria]

## 4.5 Comparações entre as Técnicas

Após a realização dos experimentos, faz-se necessário comparar as técnicas empregadas e, por meio de testes estatísticos, inferir qual delas apresentou maior precisão na construção da curva de potência de turbinas eólicas.

O apêndice A apresenta as imagens dos resultados (*p-values*) provenientes da aplicação do teste da Soma dos Postos com Sinais de Wilcoxon para Pares Combinados. O apêndice B, por sua vez, demonstra o *script* executado no *software* R para a realização da análise estatística.

Cada técnica foi representada nos testes estatísticos pelo experimento que obteve o menor EPMA, levando-se em conta o conjunto de testes aleatório. Tal experimento encontra-se marcado de azul-claro nas tabelas de apresentação do EPMA exibidas anteriormente.

O primeiro teste estatístico foi realizado na comparação entre os resultados do polinômio de grau 2 e da função sigmóide. Portanto, as hipóteses nula e alternativa foram:

$$\begin{cases} H_0 : \text{O polinômio de } 2^{\text{o}} \text{ Grau e a função sigmóide apresentam mediana dos erros estatisticamente iguais.} \\ H_1 : \text{O polinômio de } 2^{\text{o}} \text{ Grau e a função sigmóide não apresentam mediana dos erros estatisticamente iguais.} \end{cases}$$

O *p-value* calculado na realização deste teste estatístico foi bastante inferior ao nível de significância, o que ocasiona a rejeição da hipótese nula. A hipótese alternativa é aceita e, já que o EPMA proveniente da aplicação do polinômio é inferior ao da função sigmóide, conclui-se que o polinômio de grau 2 apresenta um melhor desempenho.

Posteriormente, realizou-se a comparação entre os resultados obtidos das redes MLP e RBF. As hipóteses formuladas foram:

$$\begin{cases} H_0 : \text{A rede MLP e a RBF apresentam mediana dos erros estatisticamente iguais.} \\ H_1 : \text{A rede MLP e a RBF não apresentam mediana dos erros estatisticamente iguais.} \end{cases}$$

Ao realizar este teste, obteve-se um *p-value* superior ao nível de significância, o que implica a igualdade estatística entre as duas técnicas. Portanto, a hipótese nula foi aceita e, conseqüentemente, as redes MLP e RBF apresentam desempenho estatisticamente iguais.

Por fim, realizou-se a comparação entre o polinômio, o qual apresenta desempenho superior à função sigmóide, e a rede MLP, a qual, apesar de ser estatisticamente igual à RBF, possui um tempo de execução mais rápido. As hipóteses foram:

$$\begin{cases} H_0 : \text{A rede MLP e o polinômio de } 2^{\text{o}} \text{ grau apresentam mediana dos erros estatisticamente iguais.} \\ H_1 : \text{A rede MLP e o polinômio de } 2^{\text{o}} \text{ grau não apresentam mediana dos erros estatisticamente iguais.} \end{cases}$$

O *p-value* obtido foi bastante inferior ao nível de significância, logo, a hipótese nula foi refutada. Conclui-se, portanto, que a rede MLP apresenta melhor desempenho do que o polinômio de grau 2 já que o seu EPMA foi menor.

## 5 *Considerações Finais*

### 5.1 Conclusões

É global a preocupação em não instabilizar o meio ambiente a partir da geração de energia elétrica. Em meio aos esforços pela utilização de fontes energéticas limpas e renováveis, a energia eólica surge como uma matriz de baixo custo financeiro e pequenos impactos ambientais.

Uma vez que a energia eólica conquista cada vez mais espaço no cenário mundial, este trabalho pretendeu prestar um auxílio para o planejamento de parques eólicos. Para tanto, o principal objetivo almejado foi realizar a construção de curvas de potência utilizando Redes Neurais Artificiais e confrontar os resultados obtidos com os de outras técnicas. Além disso, visou-se analisar a influência da direção do vento e dos valores anteriores de sua velocidade na construção da curva.

As redes escolhidas neste trabalho foram a MLP e a RBF. Seus resultados foram comparados com os provenientes da aplicação de um polinômio de 2<sup>o</sup> grau e de uma função sigmóide.

A fim de alcançar os objetivos, foram realizados experimentos implementados no *software* MATLAB e no R a partir de uma base de dados de um parque eólico real situado no Brasil.

Realizados os experimentos, conclui-se, primeiramente, que a não aleatoriedade dos dados que compõem o conjunto destinado aos testes conduziria a uma avaliação tendenciosa das técnicas empregadas na construção da curva já que os dados escolhidos não eram estatisticamente representativos. Isto é observado pela diferença existente entre o EPMA obtido a partir do conjunto de testes fixo e o EPMA calculado nos experimentos em que o conjunto de teste foi composto por dados selecionados aleatoriamente.

Ao comparar os resultados obtidos da aplicação das técnicas, conclui-se que a curva de potência mais precisa foi construída pela rede MLP do experimento 8. Foi provado estatisticamente, porém, que a rede MLP e a RBF possuem a mesma precisão na cons-

trução da curva. Entretanto, a rede MLP apresenta-se como melhor técnica por possuir o tempo de execução inferior ao tempo da RBF.

Em relação à adição de novas entradas nas RNAs como a direção do vento e valores anteriores da velocidade, conclui-se que não há ganho significativo de precisão nos resultados obtidos.

Por fim, algumas lacunas existentes no Guia do Usuário para o *toolbox* de RNAs do MATLAB proporcionaram dificuldades acerca de como realizar ajustes finos nas redes. Entretanto, com o auxílio de fóruns de discussão na internet e de testes empíricos, o problema foi contornado.

## 5.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se empregar as técnicas aqui abordadas para a construção de curvas de potência com dados de outros parques eólicos, a fim de consolidar o conhecimento do desempenho de RNAs neste cenário.

Almeja-se, também, realizar experimentos que envolvam a utilização de RNAs recorrentes, as quais apresentam a propriedade de armazenar os estados anteriores dos neurônios.

Por fim, em estudos futuros sobre a previsão da potência em parques eólicos serão utilizadas outras métricas para a avaliação dos resultados previstos e desejados, tais como: potência média gerada, desvio padrão da potência gerada, erro médio quadrático, erro médio absoluto, entre outras.

## *Referências*

- CERQUEIRA, W. F. *Combustíveis Fósseis*. 2008. Disponível em: <<http://www.brasilecola.com/geografia/combustiveis-fosseis.htm>>. Acesso em: 24 de setembro de 2013.
- DUTRA, R. O. *Energia Eólica: princípios e tecnologias*. [S.l.], 2008.
- FREITAS, E. *Efeito Estufa*. 2008. Disponível em: <<http://www.brasilecola.com/geografia/efeito-estufa.htm>>. Acesso em: 23 de setembro de 2013.
- JANNUZZI, G. M. S. *Além de grandes hidrelétricas: políticas para fontes renováveis de energia elétrica no Brasil*. Curitiba. Estudo publicado no 8º Congresso Brasileiro de Planejamento Energético, 2012.
- KOVÁCS, Z. L. *Redes neurais artificiais: fundamentos e aplicações*. [S.l.]: Livraria da Física, 2006.
- LI, S. et al. Comparative Analysis of Regression and Artificial Neural Network Models for Wind Turbine Power Curve Estimation. *Journal of Solar Energy Engineering*, v. 123, n. 4, p. 327, 2001. ISSN 01996231. Disponível em: <<http://solarenergyengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1456314>>.
- LUCIANA. *Mais barata e de menor impacto, energia eólica registra expansão mundial*. 2013. Disponível em: <<http://www.cidadessustentaveis.org.br/noticias/mais-barata-e-de-menor-impacto-energia-eolica-registra-expansao-mundial>>. Acesso em: 26 de setembro de 2013.
- O GLOBO. *Petróleo continuará a dominar a matriz energética mundial nas próximas décadas*. 2013. Disponível em: <<http://oglobo.globo.com/projetos/painelfgv/mat/mat14.asp>>. Acesso em: 24 de setembro de 2013.
- RABUNAL, J. R.; DORADO, J. *Artificial Neural Networks in Real-Life Applications*. [S.l.]: Idea Group Publishing, 2006.
- ROSSI, P. H. J.; OLIVEIRA, C. P. *Perguntas Frequentes sobre Energia Eólica*. 2002. Disponível em: <<http://www.pucrs.br/ce-eolica>>. Acesso em: 01 de outubro de 2013.
- SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. *Redes Neurais Artificiais: para engenharia e ciências aplicadas*. [S.l.]: Artliber, 2010.

- 
- TAVARES, M. *Apagão provoca prejuízos de R\$ 385 milhões no Nordeste*. 2013. Disponível em: <<http://oglobo.globo.com/economia/apagao-provoca-prejuizos-de-385-milhoes-no-nordeste-9752363>>. Acesso em: 09 de setembro de 2013.
- Valença, M. *Fundamentos de Redes Neurais*. [S.l.]: Livro Rápido, 2010.
- WAN, Y.; ELA, E.; ORWIG, K. *Development of an Equivalent Wind Plant Power-Curve*. Dallas, Texas. Wind Power Conference, pág. 3, 2010.

## *APÊNDICE A – Resultados dos Testes Estatísticos Realizados no Software R*

Neste apêndice são exibidas imagens dos resultados obtidos a partir da aplicação do teste da Soma dos Postos com Sinais de Wilcoxon para Pares Combinados. Nas imagens pode-se observar o *p-value* obtido nas comparações descritas na seção Comparações entre as Técnicas do capítulo 4.

```
> wilcox.test(resultados_polinomio[,1],resultados_sigmoide[,1], paired=TRUE)

Wilcoxon signed rank test

data: resultados_polinomio[, 1] and resultados_sigmoide[, 1]
V = 16, p-value = 3.148e-07
alternative hypothesis: true location shift is not equal to 0
```

Figura 11: Análise Estatística entre os Resultados do Polinômio de 2<sup>o</sup> Grau e os da Função Sigmóide.

[Fonte: elaboração própria]

```
> wilcox.test(resultados_mlp[,1],resultados_rbf[,1], paired=TRUE)

Wilcoxon signed rank test

data: resultados_mlp[, 1] and resultados_rbf[, 1]
V = 149, p-value = 0.08794
alternative hypothesis: true location shift is not equal to 0
```

Figura 12: Análise Estatística entre os Resultados da rede MLP e os da RBF.

[Fonte: elaboração própria]

```
> wilcox.test(resultados_mlp[,1],resultados_polinomio[,1], paired=TRUE)

Wilcoxon signed rank test

data: resultados_mlp[, 1] and resultados_polinomio[, 1]
V = 18, p-value = 4.712e-07
alternative hypothesis: true location shift is not equal to 0
```

Figura 13: Análise Estatística entre os Resultados da rede MLP e os do Polinômio.

[Fonte: elaboração própria]

## *APÊNDICE B - Script dos Testes Estatísticos Realizados no Software R*

```
resultados_polinomio <- read.csv("resultados_polinomio.csv")
resultados_sigmoide <- read.csv("resultados_sigmoide.csv")
resultados_mlp <- read.csv("resultados_mlp.csv")
resultados_rbf <- read.csv("resultados_rbf.csv")

wilcox.test(resultados_polinomio[,1],resultados_sigmoide[,1], paired=TRUE)

wilcox.test(resultados_mlp[,1],resultados_rbf[,1], paired=TRUE)

wilcox.test(resultados_mlp[,1],resultados_polinomio[,1], paired=TRUE)
```